# Algorithm

## User Login Authentication

1. Start
2. For user input ask user to enter email and password.
3. Validation of email format:
   1. If the email format is invalid (e.g., missing "@" or domain):
      1. Display an error message: "Invalid email format."
      2. Return to step 2 to prompt for email and password again.
4. Writing a query in database
   1. Now we will execute the query from the database so that we can retrieve the user details using the provided email.
      1. User= database.query(“select \* from users where email = email”)
5. Checking if User exists or not
   1. If user is NULL (user not found):
      1. Display an error message: "User not found."
      2. Return to step 2 to prompt for email and password again.
6. Verify Password:
   1. Compare the provided password with the stored hashed password:
      1. If user.password is equal to the hash of the provided password:
         1. Set the user session as authenticated (e.g., session.setAuthenticated(user)).
         2. Redirect to the user dashboard (e.g., redirect to dashboard).
      2. Else
         1. Display an error message: "Invalid password."
         2. Return to step 2 to prompt for email and password again.
7. End

# Pseudocode

begin

// step 1: prompt user for input

display "enter your email:"

input email

display "enter your password:"

input password

// step 2: validate email format

if email does not contain "@" or a valid domain then

display "invalid email format."

return to step 1

endif

// step 3: query the database for the user

user = query "select \* from users where email = email"

// step 4: check if user exists

if user is null then

display "user not found."

return to step 1

endif

// step 5: verify the password

if user.password == hash(password) then

// password is correct, set session as authenticated

set session.authenticated = true

display "login successful. redirecting to dashboard..."

redirect to "dashboard"

else

// password is incorrect

display "invalid password."

return to step 1

endif

end